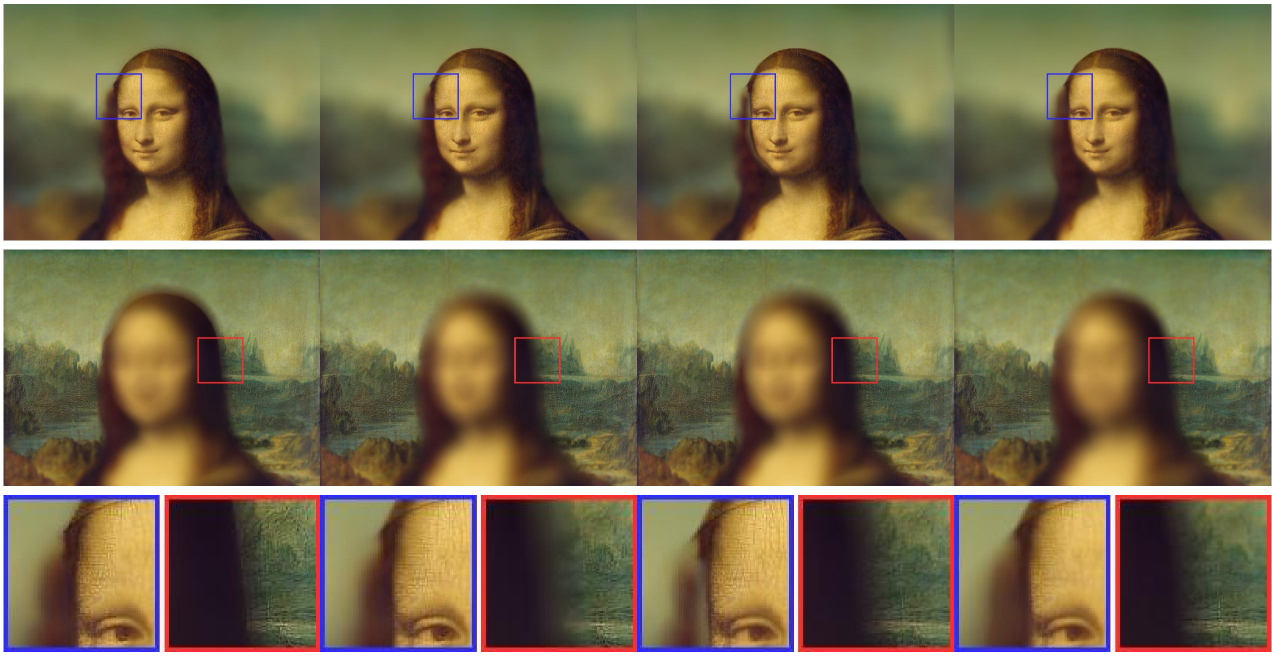
Dr.Bokeh: DiffeRentiable Occlusion-aware Bokeh Rendering

Yichen Sheng[1] Zixun Yu[2] Lu Ling[1] Zhiwen Cao[3] Xuaner Zhang[3]
Xin Lu[4] Ke Xian[5] Haiting Lin[3] Bedrich Benes[1]
[1] Purdue University [2] Google [3] Adobe [4] Typeface
[5] Huazhong University of Science and Technology



(a) SteReFo [2] (b) BokehMe [40] (c) MPIB [41] (d) Dr.Bokeh (ours)

Figure 1. Being occlusion-aware, Dr.Bokeh renders realistic bokeh effects from the bokeh rendering process without post-processing. Compared with the scattering/gathering-based method SteReFo and learning-based method BokehMe, Dr.Bokeh renders natural partial occlusion (red parts). MPIB learns to render a partial occlusion effect but breaks on unseen data (blue parts). Dr.Bokeh is more robust than learning-based methods given the same inputs because the rendering process is physically grounded. **Best viewed by zooming in**.

Abstract

Bokeh is widely used in photography to draw attention to the subject while effectively isolating distractions in the background. Computational methods can simulate bokeh effects without relying on a physical camera lens, but the inaccurate lens modeling in existing filtering-based methods leads to artifacts that need post-processing or learning-based methods to fix. We propose Dr.Bokeh, a novel rendering method that addresses the issue by directly correcting the defect that violates physics in the current filtering-based bokeh rendering equation. Dr.Bokeh first preprocesses the input RGBD to obtain a layered scene representation. Dr.Bokeh then takes the layered representation and user-defined lens parameters to render photo-realistic lens blur based on the novel occlusion-aware bokeh rendering method. Experiments show that the ***non-learning*** *based renderer Dr.Bokeh outperforms state-of-the-art bokeh rendering algorithms in terms of photo-realism. In addition, extensive quantitative and qualitative evaluations show that the more accurate lens model pushes the limit of depth-from-defocus.*

1. Introduction

Bokeh is a physical effect produced by a camera lens system. It refers to the shape and quality of out-of-focus areas in an image. Bokeh brings focus to the in-focus subject and

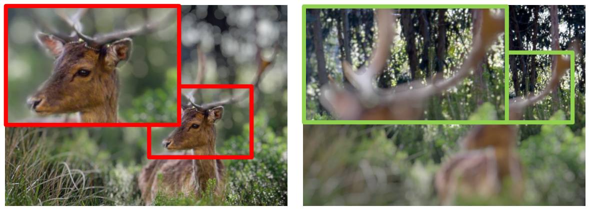

(a). Color bleeding (b). Partial occlusion

Figure 2. **Artifacts by inaccurate lens model:** color bleeding and partial occlusion are two main artifacts introduced by current inaccurate lens model. Color bleeding means the pixels in the out-of-focus scatter to in-focus regions. Partial occlusion is a semi-transparent effect on the out-of-focus boundary regions, where part of the backgrounds are visible in the background in-focus case. **Best viewed by zoom-in.**

enhances the overall aesthetic quality of the image.

Various computational methods have been developed to create the bokeh effect from an all-in-focus photo. They can be categorized as classical and learning-based methods. Classical bokeh rendering methods take an all-in-focus RGBD image as input and perform scattering or gathering operations to simulate light propagation through a thin-lens model. This simulation only considers the area where each pixel should be scattered or gathered. However, the light propagation may be occluded by other scene geometries. Failure to simulate occlusion may lead to color bleeding and unnatural partial occlusion artifacts as shown in Fig. 2. Several methods [22, 60, 76] propose to fix the color bleeding problem by compositing the all-in-focus regions back to the blur image. However, these methods involve additional steps and leave the filtering-based method incomplete and thus violate physics. Some learning-based methods [40, 61, 66] learn to render bokeh effects by training on artifact-free data. However, all the aforementioned methods fail to render partial occlusion effects naturally (Fig. 2 (b)) when the background is in focus. MPIB [41] proposes to address the problem by data-driven method. Since bokeh is a physical phenomenon, we aim to fundamentally address the challenges by proposing a more accurate lens model which physically simulates the light transportation through the lens but is grounded on a filtering-based method framework.

To achieve our goal, we introduce Dr.Bokeh, a novel occlusion-aware bokeh rendering algorithm in a filtering-based rendering process. We observe that occlusion is the key missing piece in prior works. Different from classical bokeh rendering in computer graphics that has full 3D geometry, handling occlusion in single image bokeh synthesis is much harder. Dr.Bokeh categorizes the occlusions into two types: on-focal occlusion and non-focal occlusion. Correctly simulating the two occlusions enables photorealistic rendering that is free of boundary artifacts. Dr.Bokeh does not need to be trained and can replace the bokeh renderer in existing pipelines. Note that although the rendering process is not learning-based, the input information for Dr.Bokeh, e.g., depth and inpainted background, is needed and commonly acquired by learning-based methods.

To evaluate our method, we create a synthetic benchmark where the bokeh ground truth is generated by a physically-based ray-tracer. Quantitative results on this benchmark show Dr.Bokeh significantly outperforms all the SOTA methods. A perceptual user study also shows Dr.Bokeh achieves the best visual quality. To further evaluate the proposed lens model, we make Dr.Bokeh differentiable and evaluate it on the critical application of depth-from-defocus. Experiments show the occlusion-aware rendering process outperforms the existing methods and learns the best quality depth both quantitatively and qualitatively. To summarize, our contributions are:

- Dr.Bokeh, a novel occlusion-aware filtering-based bokeh renderer by introducing geometric occlusion terms. It extends existing filtering-based methods to be more physically accurate.
- A plug-and-play differentiable Dr.Bokeh implementation which pushes the limit of depth-from-defocus field.

2. Related Work

Lens Blur Rendering Existing methods of modeling lens blur can be categorized into 3D rendering, image space filtering-based rendering and learning-based methods.

Lens blur in 3D: Ray-tracing [42–44] renders bokeh that is physically accurate. Real-time methods [5, 7, 20, 45] for efficient DOF rendering use point spread functions [23–26, 64, 68, 69] on layered representations, on view-dependent surfaces [24], lens aberration effects [25, 64], or challenging dynamic scenes [17]. Light field [59] or multiview images [33] render natural partial occlusion effects. However, a 3D scene is not always available, and it is computationally expensive to render a fully converged rendering as the camera sampling space increases.

Image-space lens blur: applies a depth-dependent blur given in-focus pixels. Classical methods [22, 60, 71, 76] use an RGBD image and kernel scattering or gathering operations, which often result in color bleeding. Later methods [2, 60, 73] propose a fix to the boundary errors by carefully blending the blurred layers. We propose a light transport simulation that naturally avoids color bleeding and produces realistic partial occlusion effects.

Learning-based methods: Recently generative methods [10, 12] and neural renderers [46–48] are used for general rendering. As for lens blur, light-field lens blur rendering [18, 53] predicts scene depth and constructs the 4D light field by warping the all-in-focus image using the depth. Other methods use differentiability of the gathering operation to learn a layered representation and render defocus blur [1, 2, 35, 53]. These approaches assume each layer has a single depth value and apply a fixed blur kernel per layer. Other methods [13, 36, 41, 61, 66] directly produce a defocused image using deep neural networks. Peng et al. [40] uses a classical method to render lens blur and then a neural network to fix the artifacts. Our algorithm can directly render realistically defocused images without any post-processing fix. Peng et al. [41] inpaints the occluded background and apply adaptive gathering operations on the multiplane image (MPI) [58] layers to make the network learn shallow depth-of-field rendering on multiple focal planes. Our blur rendering process does not need to be trained, which makes Dr.Bokeh more robust than learning-based methods. Also, Dr.Bokeh is differentiable and can be directly plugged into classical lens blur or data-driven pipelines.

Differentiable Rendering makes the rendering process suitable for inverse problems [16, 28, 32, 74, 75, 77]. Instead of handcrafting the rendering equations, others [4, 34] leverage the neural renderer directly [21, 56]. Existing methods for differentiable lens blur rendering [2, 19, 41, 53] rely on the light field or adaptive gathering operators on discrete depth layers. Our method can directly replace the blur rendering modules in those methods and output continuous depth. Gur and Wolf [8] propose a differentiable scattering-based bokeh rendering layer with a Gaussian blur kernel to learn continuous depth estimation. Dr.Bokeh follows physics laws and achieves better depth estimation results.

Image Inpainting Compared with traditional methods [3, 9], CNN-based methods [15, 39, 70] supervised by a GAN loss [6] generate plausible contents using the spatial context. Different network architectures [31, 37, 78] have been extended by different convolutions [30, 55, 72] to deal with free-form inpainting mask. Diffusion-based methods [11, 50, 51] can be used for inpainting but fail to run in real-time. We directly utilize the off-the-shelf inpainting method [55] to fill in occluded contents.

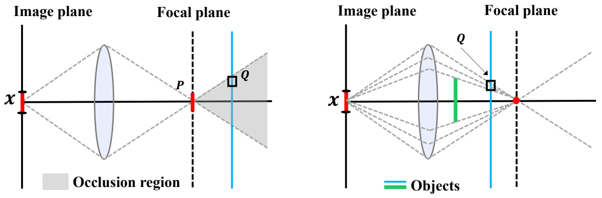
3. Image Space Bokeh Rendering

3.1. Problem Analysis

The existing filtering-based methods mainly build on the thin-lens equation that calculates the circle of confusion (Coc) area using the following equation:

$$k = \alpha\ L\ f \left| \frac{1}{z_p} - \frac{1}{z_f} \right|, \tag{1}$$

where f is focal length, α is a configurable scaling factor, L is the lens size, z_f is the focal plane depth, and z_p is the pixel depth. Filtering-based methods simulate light propagation by scattering each pixel according to Coc. However, these simulations are inaccurate when occlusion happens. Some pixels that can scatter to the neighborhood based on the Eqn. 1 may be occluded during the real light propagation process, as shown in Fig. 3. We classify the occlusions into two types: on-focal and non-focal occlusion. Without considering the on-focal occlusion, pixels at the depth boundary may incorrectly contribute to some in-focus pixels, leading to the color bleeding problem, as shown in Fig. 2 (a). The lack of non-focal occlusion modeling in the rendering leads to unnatural semi-transparent effects as shown in Fig. 2 (b).



(a). On-focal occlusion

(b). Non-focal occlusion

Figure 3. **Two occlusion problems**. Left image: In classical filtering-based method, every radiance in the cone regions is considered to contribute to the sensor x. However, the radiance inside the cone region should be occluded by the red object P on the focal plane during light propagation. Right image: although the green object occludes some of the cone regions, part of the blue object still contributes to the final results although it cannot be seen from sensor x under the pinhole camera model.

3.2. Layered Occlusion-aware Bokeh Rendering

To address the problems above, we propose to extend the existing filtering-based method to be aware of the occlusion. Similar to image-based rendering [41, 49, 57], we approximate the 3D scene by layers of RGBA images with depth. Given the layered inputs, we propose a layered occlusion-aware bokeh rendering equation:

$$B_l(x) = \sum_{l=1}^{n} V_l(x) \Pi_{k=1}^{l-1} (1 - V_k(x)) \frac{\sum_{y \in \Omega} I_l(y) w_l(y,x) O_l(y,x)}{\sum_{y \in \Omega} w_l(y,x) O_l(y,x)}, \tag{2}$$

where $O_l(y)$ and $V_l(x)$ are the on-focal occlusion and non-focal occlusion, and n is the total amount of layers. The equation computes the scattering results with on-focal occlusion for each layer, then blends the layers with non-focal occlusion terms from front to back according to the non-focal occlusion. The on-focal blending weights sum up to one to ensure energy conservation. This approach corrects the filtering-based methods by enabling spatially varying defocus blur with continuous change of blur radius and handles correctly both on-focal occlusion and non-focal occlusion at the discontinuity boundaries.

The on-focal occlusion term $O_l(y)$ ensures each layer has the correct on-focal occlusion to avoid any color bleeding. The non-focal occlusion term $V_l(x)$ ensures the correct handling of non-focal occlusion and blending of different layers, resulting in natural partial occlusion effects. In

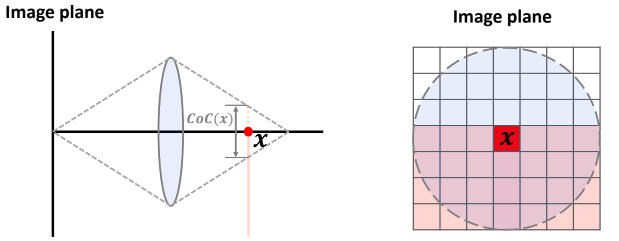


Figure 4. **Non-focal Visibility**. Left: given an image plane, a thin lens, and a rectangular-shaped pink object in the scene from a side view. Right: the image plane view of x and its neighbors. The cone region at x is visualized by the blue circle. The pink regions are the pink objects. The non-focal occlusion is computed by integrating the occlusion areas over the blue regions at x and is used to occlude the radiance contributions behind.

detail, the $V_l(x)$ term decides the occlusion percentage of the radiance behind and also the amount of energy coming from layer l. In Fig. 4, $V_l(x)$ for the pink rectangle layer reweights the energy from the rectangle layer scattering and the energy coming from layers behind such that their coefficients sum up to be one, which ensures energy conservation.

The **on-focal occlusion**, denoted by $O_l(y, x) \in \{0, 1\}$, is a unitless binary value that describes the on-focal occlusion between the neighborhood pixels y and x:

$$O_l(y,x) = \begin{cases} 0, & \text{if } d_x = 0 \text{ and } d_x > d_y \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

where d_x is the relative disparity (inverse of depth): $d_x = 1/z_x - 1/z_f$. The on-focal occlusion only happens when the radiance starting behind the object and hits objects on the focal plane. In practice, we model $O_l(y, x)$ as a probability instead of a binary value. There are several advantages: 1) The continuous value $O_l(y, x)$ better models the real physics as points not exactly on the focal plane but near the focal plane should partially occlude some amount of radiance from behind; 2) The probability is a "soft" value that models a smooth boundary occlusion. 3) The softened $O_l(y, x)$ is differentiable as explained in the **supplementary materials**.

The **non-focal occlusion** denoted by $V_l(x) \in [0, 1]$ is a unitless continuous value that describes the non-focal *visibility* between layer l and all the layers behind:

$$V_l(x) = \frac{1}{A_{\Omega'}} \sum_{y\in\Omega'} a_l(y), \quad (4)$$

where Ω' is the set of all the neighborhood pixels within the circle of confusion (CoC) region for x, $A_{\Omega'}$ is the area of Ω' and a_l is the alpha value for the layer l. The energy term $w_l(y, x)$ for the pixel y in layer l is:

$$w_l(y,x) = \frac{S_l(y,x)K(y)a(y)}{A_l(y)} = \frac{\mathbb{1}\left(\|y-x\| < \min(r,k)\right)a(y)}{\pi r^2}, \quad (5)$$

where $a(y)$ is the alpha value of y, r is the scatter radius, k is the lens size and $A_l(y)$ is the area of the CoC of y. The norm is L2-norm measuring the Euclidean distance in image space, and w is similar to [23] as it considers the energy attenuation for different scatter radius r. We further model the lens shape term K to support stylized lens shape. By default, K is a perfect circle as described in Eqn. (5). Note, w is spatially varying for different x leading to anisotropic effects.

4. Differentiable Bokeh Rendering

4.1. Soften the Non-differentiable Operations

Following SteReFo [2], Dr.Bokeh is fully differentiable by softening all the non-differentiable operations, e.g., step function or Dirac delta functions (see the **supplementary materials** for details).

4.2. Derivatives of Dr.Bokeh

The current machine learning frameworks like Pytorch [38] provide automatic differentiation mechanisms for basic mathematical operations. Dr.Bokeh cannot be directly implemented using provided auto-differentiable layers and needs custom forward/backward calculation in the CUDA layer. However, implementing the derivatives is non-trivial. We only show the derivatives w.r.t. disparity in this section. Please refer to the supplementary materials for detailed derivations of all the derivatives. For simplicity, we only derive the partial derivative for a single layer, which is enough for the implementation. Multiple layers can be easily derived based on the per-layer partial derivatives. The partial derivatives of each RGB channel are similar.

The partial derivative for d is:

$$\frac{\partial L}{\partial B(x)}\frac{\partial B(x)}{\partial d(x)} = \sum_{y\in\Omega}\frac{\partial L}{\partial B(y)}\frac{I(y)(W(y)-w(y,x)O(y,x))}{W(y)^2} \cdot \left(\frac{\partial w(y,x)}{\partial d(x)}O(y,x) + \frac{\partial O(y,x)}{\partial d(x)}w(y,x)\right), \quad (6)$$

Note that $w(y, x)$ is the energy term for pixel y to scatter to x, and $w(x, y)$ is the energy term for x to scatter to y. So $w(y, x)$ is not equivalent to $w(x, y)$. Similar to w, $O(y, x)$ is also not symmetric. Other terms can be found in supplementary materials.

4.3. Depth from Defocus

Depth from defocus is a depth estimation method that utilizes the correlation between depth and defocuses blur to train a neural network to estimate depth supervised by blur data [8, 53]. Dr.Bokeh can replace the bokeh rendering module in the previous methods and achieves better depth quality.

Training the neural network to learn depth needs a special loss function design, and we propose this loss:

$$L(y,\hat{y}) = \lambda_1 L_1(y,\hat{y}) + \lambda_2 G(\hat{y}) + \lambda_3 H_{\text{SSIM}}(y,\hat{y}), \quad (7)$$

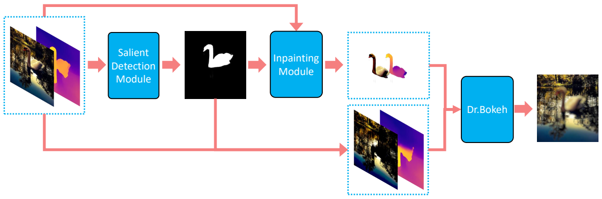


Figure 5. **Dr.Bokeh rendering pipeline** takes the RGBD image and extracts the salient object. Then, the pipeline computes the occluded RGBD values behind the salient objects. Based onthe foreground RGBAD and background RGBD, Dr.Bokeh renders a realistic bokeh image.

where λ_i are coefficients, L_1 is the norm, G is a regularization term in gradient space for smoothness [8, 53], and H_{SSIM} is the hierarchy SSIM loss to supervise Dr.Bokeh to learn a better depth. We noticed that per-pixel loss, such as $L1$, cannot supervise the neural network efficiently due to the ambiguity introduced by the bokeh computation process. Scattering or gathering is a patch-level operation. The per-pixel loss signal cannot describe the patch-level error, thus we propose adding a hierarchy SSIM term to guide the network jointly care about per-pixel results and patch results. Please refer to **supplementary materials** for more discussion.

5. Implementation and Results

Since implementing the backward differential computation is non-trivial, we will release our code to allow for the reproducibility of our work. Here we introduce our forward bokeh rendering pipeline and relevant implementation details in Sec. 5.1. Quantitative and qualitative evaluations of bokeh rendering and differentiability are discussed in Sec. 5.2, 5.3 and 5.4.

5.1. Implementation

Forward-Bokeh Rendering: For a $W \times H$ pixel image with L layers and searching neighborhood range R, the computation complexity for Dr.Bokeh is $\mathcal{O}(W \times H \times L \times R^2)$, which is the same as the classical filtering-based method. We implement Dr.Bokeh with CUDA acceleration and integrate it in Pytorch as a new computation layer. A larger L improves the bokeh rendering quality, but qualitative results in Sec. 5.3 and supplementary materials show that two layers are enough for many real-world cases. Our bokeh rendering pipeline (Fig. 5) includes an object detection module, an inpainting module, and the Dr.Bokeh renderer. An off-the-shelf salient object detection model [62] provides the layered scene representation for all examples in this paper. Please note the salient object detection module can be replaced with any other segmentation or matting network depending on the input category for good performance, e.g., an image matting network that predicts a detailed matting layer for portrait images. The salient object mask is then used to guide background RGBD inpainting, and we use LaMa [55] for high-resolution inpainting to generate all the results. Dr.Bokeh takes user-defined camera parameters, including the focal plane distance, blur radius, and lens shape, to synthesize the bokeh image.

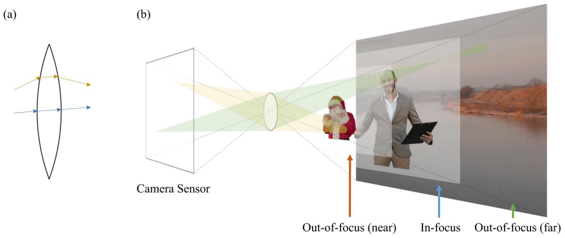


Figure 6. We simulate bokeh by computing how rays scatter and focus through a spherical lens system. (a) Side-view of how rays get refracted into and out of the lens. (b) Setup of the rendering scene. The billboards are resized to cover the exact FOV of the camera sensor.

	RMSE ↓	RMSE-s ↓	PSNR ↑	SSIM ↑	ZNCC ↑
SteReFo	0.0179	0.0178	35.92	0.9753	0.9966
DeepLens	0.0461	0.0403	27.35	0.9476	0.9827
BokehMe	0.0144	0.0143	37.77	0.9708	0.9976
MPIB	0.0152	0.0151	37.20	0.9702	0.9974
Dr.Bokeh	0.0133	0.0133	38.73	0.9757	0.9979

Table 1. Result on the synthetic benchmark. Comparing with SteReFo [2], DeepLens [61], BokehMe [40], and MPIB [41]. Dr.Bokeh outperforms state-of-the-art methods in all the metrics.

Backward-Derivatives: We implement the backward computation from scratch in CUDA and integrate the computation layer as a new layer in Pytorch. Following the Aperture [53] and GaussPSF [8], we train a CNN to predict the depth. The only difference is that we replace the bokeh rendering module with Dr.Bokeh.

5.2. Rendering Evaluation on Synthetic Benchmark

Quantitatively evaluating lens blur quality is still challenging given a single RGB or RGBD image as input and no benchmark exists yet. So we follow existing works [40, 41, 61] to create a synthetic benchmark except that we render high-quality lens blur results by ray tracing through a physically-based thin-lens.

Dataset: Existing works [41, 61] setup the scene by compositing multiple layered images and utilizing an approximated pseudo ray tracer to render the lens blur ground truth. Instead, we implemented a renderer that ray traces through a real thin lens to generate the lens blur ground truth in order to evaluate the effectiveness of Dr.Bokeh (see Fig. 6 and **supplementary** for more details). The scene (5-layer billboards) setup is similar to the dataset by DeepLens [61] and MPIB [41]. The foreground objects are randomly sampled from Adobe Matting Dataset [67] and AIM-500 [27]. The

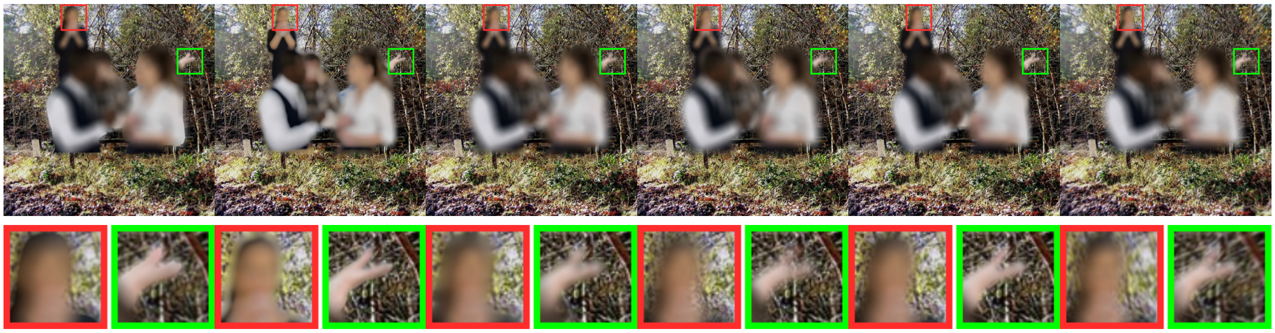


Figure 7. **Qualitative Comparison of Depth of Field (DoF) Results on Synthetic Benchmarks:** The gathering-based method (SteReFo) exhibits unnatural partial occlusion. Learning-based methods (DeepLens and BokehMe) struggle to render natural partial occlusion in the absence of explicit modeling. Although the state-of-the-art method MPIB was trained to address the partial occlusion challenge, our method Dr.Bokeh achieves the best DoF quality without necessitating training. Best viewed by zooming in.

background scenes are randomly sampled from the landmark dataset [63]. The benchmark includes 100 scenes with different blur radiuses and focal planes. Each scene has an all-in-focus image, a ground truth depth, a layered ground truth scene representation, and a bokeh ground truth.

Metric: We apply the RMSE metric and a scale-invariant RMSE (RMSE-s) [54] as we noticed that different methods have different gamma correction implementations. We also apply the SSIM and ZNCC for perception evaluation.

Comparison to related work: We compare Dr.Bokeh to SOTA methods, including a gathering-based method SteReFo [2], learning-based methods DeepLens [61], BokehMe [40] and MPIB [41]. Different methods take different kernel parameters. So we search all the blur kernels and pick the best result from each method. All methods take the same depth as input.

Each step in the lens blur rendering pipeline affects the rendering quality. But different methods have different pipelines, which makes the quantitative evaluation easy to be unfair. For example, DeepLens predicts its own depth and then predicts the lens blur. MPIB and Dr.Bokeh involves background inpainting, which looks reasonable perceptually but is easy to have large quantitative errors. To be fair to all the methods, the quantitative evaluation only measures the rendering step quality in all the pipelines instead of measuring the whole pipeline. We use the ground truth depth for all the methods. For DeepLens, we replace the predicted depth with the ground truth depth. For fairness, we replace the predicted occluded pixels with the ground truth pixels for MPIB and Dr.Bokeh.

We show quantitative and qualitative results in Tab. 1 and Fig. 7. Although the gathering-based method SteroFo achieves a high SSIM value, it is easy to observe its unnature partial occlusion results in Fig. 7 (a). DeepLens does not perform well in metrics. The potential reason is that the synthesized training data for DeepLens fails to have realistic foreground blurs. Although MPIB was designed to learn better partial occlusion effects, BokehMe still performs slightly better than MPIB in the metrics. The reason may be that MPIB is a fully learning-based method and does not generalize well to unseen datasets compared to the hybrid method BokehMe. However, MPIB qualitatively renders more natural partial occlusion effects, as shown in Fig. 7 (d). Our method Dr.Bokeh performs the best in the quantitative metrics and can render realistic partial occlusion effects.

5.3. Rendering Evaluation on Real Data

In addition to the synthetic benchmark, quantitative evaluations on lens blur are not always reliable [14, 41], so we collected real-world images and applied a user study to evaluate our method qualitatively. We also provide comprehensive qualitative results in supplementary materials.

Dataset: Quantitative results are not representative enough to draw a conclusion. Thus, we further apply a qualitative evaluation to evaluate our method. We collect real-world images with different subjects, background scenarios, and lighting as testing data for the user study. The user study contained 40 questions of two-alternative forced-choice (2AFC). Each question is a pair of lens blur results generated by Dr.Bokeh and a comparison method (SteReFo, DeepLens, BokehMe, and MPIB).

Comparison to related works: The metric and the comparing works are the same as the quantitative evaluation: 41 people (75% male, 25% female, 46% no photography experience, 26% some experience, 28% experienced) participated in our user study. We discard all replies that were too short (under three minutes) or always clicked on the same side. Our results show that 81%, 74%, 68%, 61% of participants support that the image generated by Dr.Bokeh is more realistic than SteReFo, DeepLens, BokehMe, and MPIB. In particular, the T-test value in the more realistic lens blur

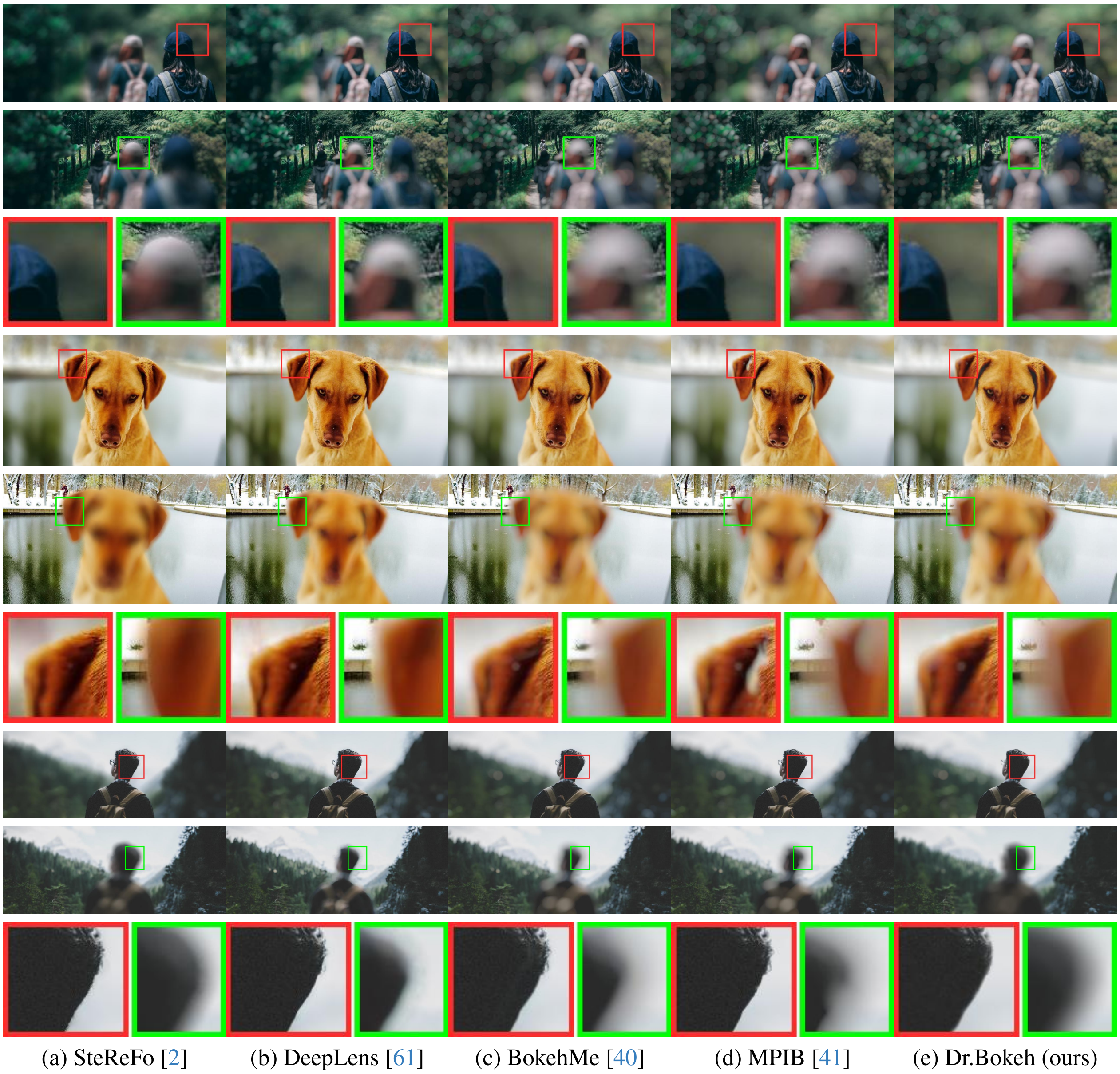


Figure 8. **Qualitative comparisons on real-world images:** Classical methods (SteReFo) are competitive for foreground in-focus but fail to render natural boundary partial occlusion. Learning-based methods (DeepLens and BokehMe) suffer from unnatural partial occlusion effects. MPIB learns to render the partial occlusion effect but has leaking artifacts due to generalization issues (see the second and the third-row examples). Dr.Bokeh renders natural partial occlusion effects and is more robust for either foreground or background in-focus cases given the same inputs.

effect for an image generated by Dr.Bokeh and BokehMe, DeepLens, and SteReFo are 5.75, 8.12, and 11.63 and are significant at 0.001 levels, which indicates the Dr.Bokeh is significantly better than those reference methods from a user perception perspective.

Fig. 8 demonstrates the qualitative comparisons with all the previous works. For foreground in-focus cases, Dr.Bokeh can preserve the sharp boundary consistently, while the learning-based methods still have generalization issues such as unnatural partial occlusion effects and leaky background artifacts (see Fig. 8 the second and third examples). For background in-focus cases, Dr.Bokeh has natural partial occlusion effects and is more robust than SOTA learning-based method MPIB. More qualitative comparison results can be found in the supplementary materials.

	RMSE ↓	SSIM ↑	PSNR ↑
Aperture	0.0133	0.9774	37.84
GaussPSF	0.0132	0.9767	37.84
Dr.Bokeh L1 + Grad	0.0146	0.9673	36.94
Dr.Bokeh L1 + Grad + SSIM	0.0136	0.9740	37.59
Dr.Bokeh w.o. occlusion	0.0139	0.9729	37.40
Dr.Bokeh	0.0123	0.9807	38.45

Table 2. Result on the light field benchmark and ablation study. Compare with aperture supervision [53] and GaussPSF [8]. Note the metrics are measured on the DoF image due to the lack of depth ground truth. Aperture, GaussPSF and Dr.Bokeh w.o. occlusion use the same loss with Dr.Bokeh. Correct handling of boundary occlusion improves the existing gathering-based depth form defocus performance. The proposed hierarchy SSIM loss further helps Dr.Bokeh learns better depth.

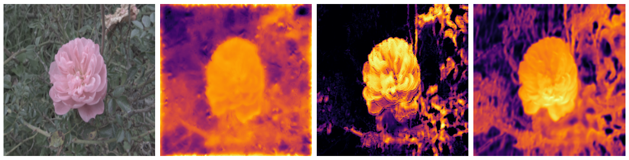


Figure 9. **Depth optimization examples**. The first row is to optimize the depth from only one pair data. The second row is the depth from light field dataset. The first column is the all-in-focus input image. The second column shows results by Aperture [53]; the third column by GaussPSF [8], and the last column results by Dr.Bokeh. The depth map optimized by Dr.Bokeh has more details and is more accurate.

5.4. Differentiable Evaluation

Comparison to Related Work: We relate our differentiable lens blur rendering to two previous works: Aperture Supervision (Aperture) [53] and Guassian-based PSF (GaussPSF)[8]. Aperture trains a neural network to predict depth layers by blur image supervision. GaussPSF replaces the bokeh rendering module in Aperture with differentiable Gaussian kernels and trains the neural network in a similar routine. Compared to GaussPSF, our occlusion-aware Dr.Bokeh is a more accurate differentiable bokeh rendering module in terms of lens blur physics. For a fair comparison in the following benchmarks, we use the same depth estimation network [65] for all the comparison methods and the same loss functions Eqn. (7) for all methods.

Benchmark: We evaluate the differentiability of Dr.Bokeh on the real-world benchmark: *Light Field Dataset* [52]. There is no depth ground truth for the real datasets, so we only quantitatively evaluate the final rendered bokeh images and qualitatively show depth qualities from all the methods. The bokeh images rendered from the light-field camera are good bokeh approximations. There are 3,343 images. Similar to previous works, we split the dataset into 3,006 training images and 337 testing images.

Depth Quality: Tab. 2 shows the quantitative evaluation results. Dr.Bokeh outperforms all the previous works in all metrics, which shows that a more accurate blur renderer improves the learning process. The quantitative evaluation is measured on bokeh images and we show the qualitative results of the generated depth map in Fig. 9. The depth map can either be obtained by direct optimization over an all-in-focus image and a bokeh image pair or by training a neural network to predict the depth based on a large-scale defocus dataset. The direct optimization over one-pair data can clearly show the depth quality supervised by the differentiable rendering layer, while the depth predicted by the trained neural network can illustrate the overall performance of the differentiable layer in the data-driven pipeline. As shown in Fig. 9, Dr.Bokeh can obtain the best quality depth image supervised by the defocus image in both settings.

Ablation Study: We conduct two experiments to understand the contribution of the occlusion term and the proposed hierarchy SSIM (HSSIM) loss (Sec. 4.2). We first compare Dr.Bokeh with a similar differentiable rendering layer but without the occlusion term by training on the *light field* benchmark. Evaluations (see Tab. 2) on the benchmark show that the occlusion term helps the neural network training. Second, in the loss function experiment, we compare our loss function (Eqn. 7) with two similar versions: one is just a $L1$ loss with the gradient loss, and the other is the $L1$ loss with the gradient loss and the SSIM loss. Tab. 2 shows our loss function with HSSIM outperforms the best. Please refer to supplementary for more qualitative ablation study results.

6. Conclusions

We have introduced Dr.Bokeh, a novel differentiable occlusion-aware DoF rendering algorithm. Dr.Bokeh addresses the color bleeding problem and renders realistic partial occlusion for DoF effect synthesis by proposing a more accurate lens model. Moreover, Dr.Bokeh is a plug-and-play differentiable DoF rendering module that can be used in data-driven pipelines. Qualitative and quantitative comparisons validate that Dr.Bokeh achieves the state-of-the-art lens blur quality in different focus settings and the state-of-the-art depth quality in the depth-from-defocus community.

Limitation and Future Work: Similar to other existing works, inaccurate image inputs (depth and inpainting) lead to artifacts. To address this, a potential future direction could involve utilizing Dr.Bokeh's differentiability to learn robustness against noisy inputs. Recent dataset DL3DV [29] may help.

7. Acknowledgement

Part of this work is done during Yichen's Adobe's internship. Ke Xian's part is supported by Guangxi Science and Technology Project (AB21196034).

References

[1] Hadi Alzayer, Abdullah Abuolaim, Leung Chun Chan, Yang Yang, Ying Chen Lou, Jia-Bin Huang, and Abhishek Kar. Dc2: Dual-camera defocus control by learning to refocus. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21488–21497, 2023. 3

[2] Benjamin Busam, Matthieu Hog, Steven McDonagh, and Gregory Slabaugh. SteReFo: Efficient Image Refocusing with Stereo Vision. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3295–3304, Seoul, Korea (South), 2019. IEEE. 1, 2, 3, 4, 5, 6, 7

[3] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Object removal by exemplar-based inpainting. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, pages II–II. IEEE, 2003. 3

[4] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360 (6394):1204–1210, 2018. 3

[5] Linus Franke, Nikolai Hofmann, Marc Stamminger, and Kai Selgrad. Multi-layer depth of field rendering with tiled splatting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):1–17, 2018. 2

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3

[7] Jhonny Göransson and Andreas Karlsson. Practical post-process depth of field. *GPU Gems*, 3(583-606):2, 2007. 2

[8] Shir Gur and Lior Wolf. Single Image Depth Estimation Trained via Depth From Defocus Cues. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7675–7684, Long Beach, CA, USA, 2019. IEEE. 3, 4, 5, 8

[9] James Hays and Alexei A Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (ToG)*, 26(3):4–es, 2007. 3

[10] Liu He and Daniel Aliaga. Globalmapper: Arbitrary-shaped urban layout generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 454–464, 2023. 2

[11] Liu He, Yijuan Lu, John Corring, Dinei Florencio, and Cha Zhang. Diffusion-based document layout generation. In *Document Analysis and Recognition - ICDAR 2023: 17th International Conference, San José, CA, USA, August 21–26, 2023, Proceedings, Part I*, page 361–378, Berlin, Heidelberg, 2023. Springer-Verlag. 3

[12] Liu He, Jie Shan, and Daniel Aliaga. Generative building feature estimation from satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–13, 2023. 2

[13] Andrey Ignatov, Jagruti Patel, and Radu Timofte. Rendering natural camera bokeh effect with deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 418–419, 2020. 3

[14] Andrey Ignatov, Radu Timofte, Ming Qian, Congyu Qiao, Jiamin Lin, Zhenyu Guo, Chenghua Li, Cong Leng, Jian Cheng, Juewen Peng, et al. Aim 2020 challenge on rendering realistic bokeh. In *European Conference on Computer Vision*, pages 213–228. Springer, 2020. 6

[15] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017. 3

[16] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. Dr. jit: a just-in-time compiler for differentiable rendering. *ACM Transactions on Graphics (TOG)*, 41(4):1–19, 2022. 3

[17] Yuna Jeong, Seung Youp Baek, Yechan Seok, Gi Beom Lee, and Sungkil Lee. Real-time dynamic bokeh rendering with efficient look-up table sampling. *IEEE Transactions on Visualization and Computer Graphics*, 28(2):1373–1384, 2020. 2

[18] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Trans. Graph.*, 35(6):1–10, 2016. 2

[19] Takuhiro Kaneko. Unsupervised Learning of Depth and Depth-of-Field Effect from Natural Images with Aperture Rendering Generative Adversarial Networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15674–15683, Nashville, TN, USA, 2021. IEEE. 3

[20] Michael Kass, Aaron Lefohn, and John D Owens. Interactive depth of field using simulated diffusion on a gpu. 2006. 2

[21] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020. 3

[22] M. Kraus and M. Strengert. Depth-of-Field Rendering by Pyramidal Image Processing. *Computer Graphics Forum*, 26(3):645–654, 2007. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2007.01088.x. 2

[23] Sungkil Lee, Gerard Jounghyun Kim, and Seungmoon Choi. Real-time depth-of-field rendering using point splatting on per-pixel layers. In *Computer Graphics Forum*, pages 1955–1962. Wiley Online Library, 2008. 2, 4

[24] Sungkil Lee, Elmar Eisemann, and Hans-Peter Seidel. Depth-of-field rendering with multiview synthesis. *ACM Transactions on Graphics (TOG)*, 28(5):1–6, 2009. 2

[25] Sungkil Lee, Elmar Eisemann, and Hans-Peter Seidel. Real-time lens blur effects and focus control. *ACM Trans. Graph.*, 29(4):1–7, 2010. 2

[26] Kefei Lei and John F Hughes. Approximate depth of field effects using few samples per pixel. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 119–128, 2013. 2

[27] Jizhizi Li, Jing Zhang, and Dacheng Tao. Deep automatic natural image matting. *arXiv preprint arXiv:2107.07235*, 2021. 5

[28] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 3

[29] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. *arXiv preprint arXiv:2312.16256*, 2023. 8

[30] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 85–100, 2018. 3

[31] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *European Conference on Computer Vision*, pages 725–741. Springer, 2020. 3

[32] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019. 3

[33] Xin Liu and Jon G Rokne. Depth of field synthesis from sparse views. *Computers & Graphics*, 55:21–32, 2016. 2

[34] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 3

[35] Xianrui Luo, Juewen Peng, Ke Xian, Zijin Wu, and Zhiguo Cao. Bokeh Rendering from Defocus Estimation. In *Computer Vision – ECCV 2020 Workshops*, pages 245–261. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science. 3

[36] Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, Hans-Peter Seidel, and Tobias Ritschel. Deep Shading: Convolutional Neural Networks for Screen-Space Shading. *Computer Graphics Forum*, 36(4):65–78, 2017. arXiv:1603.06078 [cs]. 3

[37] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 4

[39] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 3

[40] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. BokehMe: When Neural Rendering Meets Classical Rendering. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16262–16271, New Orleans, LA, USA, 2022. IEEE. 1, 2, 3, 5, 6, 7

[41] Juewen Peng, Jianming Zhang, Xianrui Luo, Hao Lu, Ke Xian, and Zhiguo Cao. MPIB: An MPI-Based Bokeh Rendering Framework for Realistic Partial Occlusion Effects. In *Computer Vision – ECCV 2022*, pages 590–607, Cham, 2022. Springer Nature Switzerland. 1, 2, 3, 5, 6, 7

[42] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016. 2

[43] Michael Potmesil and Indranil Chakravarty. A lens and aperture camera model for synthetic image generation. *ACM SIGGRAPH Computer Graphics*, 15(3):297–305, 1981.

[44] Przemyslaw Rokita. Generating depth-of-field effects in virtual reality applications. *IEEE Computer Graphics and Applications*, 16(2):18–21, 1996. 2

[45] Thorsten Scheuermann et al. Advanced depth of field. *GDC 2004*, 8, 2004. 2

[46] Yichen Sheng, Jianming Zhang, and Bedrich Benes. Ssn: Soft shadow network for image compositing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4380–4390, 2021. 2

[47] Yichen Sheng, Yifan Liu, Jianming Zhang, Wei Yin, A Cengiz Oztireli, He Zhang, Zhe Lin, Eli Shechtman, and Bedrich Benes. Controllable shadow generation using pixel height maps. In *European Conference on Computer Vision*, pages 240–256. Springer, 2022.

[48] Yichen Sheng, Jianming Zhang, Julien Philip, Yannick Hold-Geoffroy, Xin Sun, He Zhang, Lu Ling, and Bedrich Benes. Pixht-lab: Pixel height based light effect generation for image compositing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16643–16653, 2023. 2

[49] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3D Photography Using Context-Aware Layered Depth Inpainting. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8025–8035, Seattle, WA, USA, 2020. IEEE. 3

[50] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. Objectstitch: Object compositing with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18310–18319, 2023. 3

[51] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, He Zhang, Wei Xiong, and Daniel Aliaga. Imprint: Generative object compositing by learning identity-preserving representation. *arXiv preprint arXiv:2403.10701*, 2024. 3

[52] Pratul P Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. Learning to synthesize a 4d rgbd light field from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2243–2251, 2017. 8

[53] Pratul P. Srinivasan, Rahul Garg, Neal Wadhwa, Ren Ng, and Jonathan T. Barron. Aperture Supervision for Monocular Depth Estimation. In *2018 IEEE/CVF Conference on Com-

puter Vision and Pattern Recognition, pages 6393–6401, Salt Lake City, UT, 2018. IEEE. 2, 3, 4, 5, 8

[54] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Transactions on Graphics (TOG)*, 38(4):79–1, 2019. 6

[55] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2149–2159, 2022. 3, 5

[56] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*, pages 701–727. Wiley Online Library, 2020. 3

[57] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[58] Richard Tucker and Noah Snavely. Single-View View Synthesis With Multiplane Images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 548–557, Seattle, WA, USA, 2020. IEEE. 3

[59] Karthik Vaidyanathan, Jacob Munkberg, Petrik Clarberg, and Marco Salvi. Layered light field reconstruction for defocus blur. *ACM Transactions on Graphics (TOG)*, 34(2): 1–12, 2015. 2

[60] Neal Wadhwa, Rahul Garg, David E. Jacobs, Bryan E. Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T. Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Trans. Graph.*, 37(4):1–13, 2018. 2

[61] Lijun Wang, Xiaohui Shen, Jianming Zhang, Oliver Wang, Zhe Lin, Chih-Yao Hsieh, Sarah Kong, and Huchuan Lu. DeepLens: Shallow Depth Of Field From A Single Image. *arXiv:1810.08100 [cs]*, 2018. arXiv: 1810.08100. 2, 3, 5, 6, 7

[62] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13025–13034, 2020. 5

[63] T. Weyand, A. Araujo, B. Cao, and J. Sim. Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In *Proc. CVPR*, 2020. 6

[64] Jiaze Wu, Changwen Zheng, Xiaohui Hu, and Fanjiang Xu. Rendering realistic spectral bokeh due to lens stops and aberrations. *The Visual Computer*, 29:41–52, 2013. 2

[65] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 8

[66] Lei Xiao, Anton Kaplanyan, Alexander Fix, Matthew Chapman, and Douglas Lanman. DeepFocus: learned image synthesis for computational displays. *ACM Trans. Graph.*, 37 (6):1–13, 2018. 2, 3

[67] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2970–2979, 2017. 5

[68] Shibiao Xu, Xing Mei, Weiming Dong, Xun Sun, Xukun Shen, and Xiaopeng Zhang. Depth of field rendering via adaptive recursive filtering. In *SIGGRAPH Asia 2014 Technical Briefs*, pages 1–4. 2014. 2

[69] Ling-Qi Yan, Soham Uday Mehta, Ravi Ramamoorthi, and Fredo Durand. Fast 4d sheared filtering for interactive rendering of distribution effects. *ACM Transactions on Graphics (TOG)*, 35(1):1–13, 2015. 2

[70] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6721–6729, 2017. 3

[71] Yang Yang, Haiting Lin, Zhan Yu, Sylvain Paris, and Jingyi Yu. Virtual DSLR: High Quality Dynamic Depth-of-Field Synthesis on Mobile Platforms. *ei*, 28(18):1–9, 2016. 2

[72] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4471–4480, 2019. 3

[73] Benxuan Zhang, Bin Sheng, Ping Li, and Tong-Yee Lee. Depth of field rendering using multilayer-neighborhood optimization. *IEEE Transactions on Visualization and Computer Graphics*, 26(8):2546–2559, 2019. 2

[74] Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. A differential theory of radiative transfer. *ACM Transactions on Graphics (TOG)*, 38(6):1–16, 2019. 3

[75] Cheng Zhang, Bailey Miller, Kan Yan, Ioannis Gkioulekas, and Shuang Zhao. Path-space differentiable rendering. *ACM transactions on graphics*, 39(4), 2020. 3

[76] Xuaner Zhang, Kevin Matzen, Vivien Nguyen, Dillon Yao, You Zhang, and Ren Ng. Synthetic defocus and look-ahead autofocus for casual videography. *ACM Trans. Graph.*, 38 (4):1–16, 2019. 2

[77] Shuang Zhao, Wenzel Jakob, and Tzu-Mao Li. Physics-based differentiable rendering: from theory to implementation. In *ACM siggraph 2020 courses*, pages 1–30. 2020. 3

[78] Manyu Zhu, Dongliang He, Xin Li, Chao Li, Fu Li, Xiao Liu, Errui Ding, and Zhaoxiang Zhang. Image inpainting by end-to-end cascaded refinement with mask awareness. *IEEE Transactions on Image Processing*, 30:4855–4866, 2021. 3